WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, April 19, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L24	L23 and (context near (switch or swap))	13
☐	L23	L22	69
		DB=USPT,PGPB; PLUR=YES; OP=ADJ	
☐	L22	L21 and breakpoint	105
☐	L21	L20 and debug$	258
☐	L20	L18 and (thread or multithread$ or multi?thread$)	497
☐	L19	L18 and L1	12
☐	L18	L17 or L16 or L15	2143
☐	L17	((712/228 |712/229)!.CCLS.)	640
☐	L16	((709/106 |709/107 |709/108)!.CCLS.)	0
☐	L15	((717/124 |717/125 |717/126 |717/127 |717/128 |717/129 |717/130 |717/131 |717/132 |717/133 |717/158)!.CCLS.)	1508
		DB=TDBD; PLUR=YES; OP=ADJ	
☐	L14	context swap	5
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L13	L12	6
		DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L12	multithread$ and context swap	19
☐	L11	L10 and debug$	47
☐	L10	multithread$ and context switches	189
☐	L9	multithreading and context switches	103
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L8	L7 and thread	28
☐	L7	(vo, ted).xa.	130
		DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L6	L5 and debug$	54
☐	L5	L4 and swap	159
☐	L4	(thread or multithread) near2 processor	1692
☐	L3	L2 and debug$	8
☐	L2	L1 and (thread or multithread)	89
☐	L1	micro-engine or microengine	410

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 13 of 13 returned.

☐ 1. Document ID: US 6715139 B1

L24: Entry 1 of 13 File: USPT Mar 30, 2004

US-PAT-NO: 6715139
DOCUMENT-IDENTIFIER: US 6715139 B1

TITLE: System and method for providing and displaying debugging information of a graphical program on a first computer during execution of the graphical program on a second computer

DATE-ISSUED: March 30, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kodosky; Jeffrey L	Austin	TX		
Shah; Darshan	Round Rock	TX		
DeKey; Samson	Austin	TX		
Rogers; Steve	Austin	TX		

US-CL-CURRENT: 717/125; 717/124, 717/126, 717/131

ABSTRACT:

A computer-based virtual instrumentation system including a host computer and an embedded system or device, wherein graphical programs created using the computer system can be downloaded to the embedded system for execution in a real-time or more deterministic manner. The present invention thus provides a method for automatically generating an embedded application in response to a graphical program created by a user. This provides the user the ability to develop or define instrument functionality using graphical programming techniques, while enabling the resulting program to operate in an embedded real-time system. The invention includes a novel method for configuring the embedded system. During execution of a graphical program in the embedded system, the block diagram portion executes in the embedded system, and the host CPU executes front panel display code to display on the screen the graphical front panel of the graphical program. The embedded system and the host computer exchange data using a front panel protocol to enable this operation. The present invention also includes improved debugging support for graphical programs executing on the embedded system. The host graphical programming system thus provides the user interface for graphical programs executing on the embedded system, essentially acting as the front panel "browser" for embedded applications. The host LabVIEW can also act as an independent application communicating with embedded LabVIEW through the shared memory. The host graphical programming system further provides a seamless environment in which the user can develop an embedded application using high level graphical programming techniques.

50 Claims, 19 Drawing figures

Exemplary Claim Number: 1
Number of Drawing Sheets: 17

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw Desc

☐ 2. Document ID: US 6708326 B1

L24: Entry 2 of 13 File: USPT Mar 16, 2004

US-PAT-NO: 6708326
DOCUMENT-IDENTIFIER: US 6708326 B1

TITLE: Method, system and program product comprising breakpoint handling mechanism for debugging and/or monitoring a computer instruction sequence

DATE-ISSUED: March 16, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bhattacarya; Suparna	Karnataka			IN

US-CL-CURRENT: 717/124; 712/227, 714/34

ABSTRACT:

A computer method, system and program product for debugging and/or monitoring an instruction set and having an improved breakpoint handling mechanism involving a hardware debug register set (or like breakpoint register means). Instead of patching a break instruction into a debuggee instruction sequence, re-inserting the original instruction and then single stepping through that instruction before replacing it with the patch, the original instruction is left in place and continuous execution is resumed. Before resuming however, the breakpoint register is set so that the break instruction can be re-applied while a flag (eg the Intel RF flag) is set so as to prevent a hardware break before that is desired.

7 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWMC | Draw Desc

☐ 3. Document ID: US 6629123 B1

L24: Entry 3 of 13 File: USPT Sep 30, 2003

US-PAT-NO: 6629123
DOCUMENT-IDENTIFIER: US 6629123 B1

TITLE: Interception of unit creation requests by an automatic distributed partitioning system

DATE-ISSUED: September 30, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hunt; Galen C.	Bellevue	WA		

US-CL-CURRENT: 718/106; 717/131, 719/310

ABSTRACT:

An automatic distributed partitioning system (ADPS) intercepts function calls to unit activation functions that dynamically create application units, such as a component instantiation function. A system service library provides a unit activation function. An application program includes at least one function call to the unit activation function. The ADPS redirects the function call to instrumentation of the ADPS. In one technique, the ADPS uses inline redirection of the function call to the unit activation function.

23 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

☐ 4. Document ID: US 6381735 B1

L24: Entry 4 of 13 File: USPT Apr 30, 2002

US-PAT-NO: 6381735
DOCUMENT-IDENTIFIER: US 6381735 B1

TITLE: Dynamic classification of sections of software

DATE-ISSUED: April 30, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hunt; Galen C.	Bellevue	WA		

US-CL-CURRENT: 717/158

ABSTRACT:

Dynamic classification of sections of software using a profile-based optimization system optimizes management of the sections of software. Software executes under expected usage conditions. After execution, a set of usage profiles describes the dynamic properties of sections of the software. Each usage profile includes information identifying a section of software. Each usage profile maps to an outcome meant to optimize management of the sections of the software during later execution. During such later execution, a usage background describes the dynamic properties of a section of the software. The usage background includes information identifying the section of software. By matching the usage background to a usage profile in the set of usage profiles, the section is dynamically classified during later execution. Based on this dynamic classification, the section maps to the outcome meant to optimize management of the sections of software.

61 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw Desc

☐ 5. Document ID: US 6282701 B1

L24: Entry 5 of 13 File: USPT Aug 28, 2001

US-PAT-NO: 6282701
DOCUMENT-IDENTIFIER: US 6282701 B1

TITLE: System and method for monitoring and analyzing the execution of computer programs

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wygodny; Shlomo	Ramat Hasharon			IL
Barboy; Dmitry	Rehovot			IL
Prouss; Georgi	Kiev			UA
Vorobey; Anatoly	Rishon Lezion			IL

US-CL-CURRENT: 717/125; 702/183, 702/187, 709/224, 714/35, 714/38, 714/45, 714/46, 717/128, 717/163, 719/317, 719/331

ABSTRACT:

A software system is disclosed which facilitates the process of tracing the execution paths of a program, called the client. The tracing is performed without requiring modifications to the executable or source code files of the client. Trace data collected during the tracing operation is collected according to instructions in a trace options file. At run time, the tracing library attaches to the memory image of the client. The tracing library is configured to monitor execution of the client and to collect trace data, based on selections in the trace options file. The developer then uses a trace analyzer program, also having a graphical user interface, to view the trace information. The system can trace multiple threads and multiple processes. The tracing library is preferably configured to runs in the same process memory space as the client thereby tracing the execution of the client program without the need for context switches. The tracing system provides a remote mode and an online mode. In remote mode, the developer sends the trace control information to a remote user site together with a small executable image called the agent that enables a remote customer, to generate a trace file that represents execution of the client application at the remote site. In online mode, the developer can generate trace options, run and trace the client, and display the trace results in near real-time on the display screen during execution of the client program.

31 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

☐ 6. Document ID: US 6263491 B1

L24: Entry 6 of 13 File: USPT Jul 17, 2001

US-PAT-NO: 6263491
DOCUMENT-IDENTIFIER: US 6263491 B1

TITLE: Heavyweight and lightweight instrumentation

DATE-ISSUED: July 17, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hunt; Galen C.	Bellevue	WA		

US-CL-CURRENT: 717/130

ABSTRACT:

An instrumentation system performs operations such as profiling an application and partitioning and distributing units of the application using different versions of metadata describing the application. Performing an operation on an executing application generates overhead. Detailed metadata used in operations such as profiling create unnecessary overhead during other operations. By removing metadata detail unnecessary for a particular operation, an instrumentation system using reduced metadata generates less overhead for that particular operation. Different instrumentation packages include different versions of metadata for performing operations on the application.

36 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw Desc

☐ 7. Document ID: US 6260150 B1

L24: Entry 7 of 13 File: USPT Jul 10, 2001

US-PAT-NO: 6260150
DOCUMENT-IDENTIFIER: US 6260150 B1

TITLE: Foreground and background context controller setting processor to power saving mode when all contexts are inactive

DATE-ISSUED: July 10, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Diepstraten; Wilhelmus J. M.	Haghorst			NL

Fischer; Michael A.	San Antonio	TX
Hardell; Wesley D.	San Antonio	TX

US-CL-CURRENT: 713/323; 712/229, 718/107

ABSTRACT:

A context controller for managing multitasking in a processor and a method of operating the same. In one embodiment, the context controller includes: (1) foreground and background task controllers that allocate processor resources to active contexts corresponding to foreground and background tasks, respectively, and (2) mode switching circuitry, coupled to the foreground and background task controllers, that places the processor in an idle state and a power saving mode when all of the contexts are inactive.

22 Claims, 21 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 21

☐ 8. Document ID: US 6205468 B1

L24: Entry 8 of 13 File: USPT Mar 20, 2001

US-PAT-NO: 6205468
DOCUMENT-IDENTIFIER: US 6205468 B1

TITLE: System for multitasking management employing context controller having event vector selection by priority encoding of contex events

DATE-ISSUED: March 20, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Diepstraten; Wilhelmus J. M.	Haghorst			NL
Fischer; Michael A.	San Antonio	TX		
Hardell; Wesley D.	San Antonio	TX		

US-CL-CURRENT: 718/108; 712/228, 712/229, 718/100, 718/101, 718/102, 718/103, 718/104, 718/105, 718/106, 718/107

ABSTRACT:

A context controller for managing multitasking in a processor and a method of operating the same. In one embodiment, the context controller includes: (1) an event recorder that records occurrences of events and (2) an encoder, associated with the event recorder, that, in response to a software instruction, priority encodes bits corresponding to at least some of the events to generate therefrom an event-dependent vector to allow the processor to branch as a function thereof. Vectoring is per-instance of the vector decode software instruction, not per-event or per-context.

22 Claims, 21 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 21

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Drawi Desc | Ir

☐ 9. Document ID: US 6202199 B1

L24: Entry 9 of 13 File: USPT Mar 13, 2001

US-PAT-NO: 6202199
DOCUMENT-IDENTIFIER: US 6202199 B1

TITLE: System and method for remotely analyzing the execution of computer programs

DATE-ISSUED: March 13, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wygodny; Shlomo	Ramat Hasharon			IL
Barboy; Dmitry	Rehovot			IL
Prouss; Georgi	Kiev			UA
Vorobey; Anatoly	Rishon Lezion			IL

US-CL-CURRENT: 717/125; 702/183, 717/128

ABSTRACT:

A software system is disclosed which facilitates the process of tracing the execution paths of a program, called the client. The tracing is performed without requiring modifications to the executable or source code files of the client. Trace data collected during the tracing operation is collected according to instructions in a trace options file. At run time, the tracing library attaches to the memory image of the client. The tracing library is configured to monitor execution of the client and to collect trace data, based on selections in the trace options file. The developer then uses a trace analyzer program, also having a graphical user interface, to view the trace information. The system can trace multiple threads and multiple processes. The tracing library is preferably configured to runs in the same process memory space as the client thereby tracing the execution of the client program without the need for context switches. The tracing system provides a remote mode and an online mode. In remote mode, the developer sends the trace control information to a remote user site together with a small executable image called the agent that enables a remote customer, to generate a trace file that represents execution of the client application at the remote site. In online mode, the developer can generate trace options, run and trace the client, and display the trace results in near real-time on the display screen during execution of the client program.

33 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Drawi Desc | Ir

☐ 10. Document ID: US 6058465 A

L24: Entry 10 of 13 File: USPT May 2, 2000

US-PAT-NO: 6058465
DOCUMENT-IDENTIFIER: US 6058465 A

TITLE: Single-instruction-multiple-data processing in a multimedia signal processor

DATE-ISSUED: May 2, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Le Trong	Monte Sereno	CA	95030	

US-CL-CURRENT: 712/7; 708/441, 712/13, 712/228, 712/23, 712/4, 712/5

ABSTRACT:

A vector processor architecture provides vector registers of fixed size having data elements of programmable size and type. The type and size for data elements are defined by instructions which manipulate operands associated with the vector registers. The data size defined by an instruction determines the number of the data elements in a vector register and the number of parallel operations performed to complete the instruction. One embodiment of the invention supports 8-bit, 9-bit, 16-bit, and 32-bit data element sizes of integer type for all sizes and floating point data type for the 32-bit data elements.

14 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10

☐ 11. Document ID: US 6052773 A

L24: Entry 11 of 13 File: USPT Apr 18, 2000

US-PAT-NO: 6052773
DOCUMENT-IDENTIFIER: US 6052773 A

TITLE: DPGA-coupled microprocessors

DATE-ISSUED: April 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
DeHon; Andre	Cambridge	MA		
Bolotski; Michael	Somerville	MA		
Knight, Jr.; Thomas F.	Belmont	MA		

US-CL-CURRENT: 712/43; 712/229

ABSTRACT:

A single chip microprocessor or memory device has reprogrammable characteristics according to the invention. In the case of the microprocessor, a fixed processing cell is provided as is common to perform logic calculations. A portion of the chip silicon real-estate, however, is dedicated a programmable gate array. This feature enables application-specific configurations to allow adaptation to the particular time-changing demands of the microprocessor and provide the functionality required to best serve those demands. This yields application acceleration and in system-specific functions. In other cases the configurable logic acts as network interface, which allows the same basic processor design to function in any environment to which the interface can adapt.

The invention also concerns a memory device having a plurality of memory banks and configurable logic units associated with the memory banks. An interconnect is provided to enable communication between the configurable logic units. These features lessen the impact of the data bottle-neck associated with bus communications, since the processing capability is moved to the memory in the form programmable logic, which can be configured to the needs of the specific application. The inherently large on-chip bandwidth can then be utilized to increase the speed at which bulk data is processed.

22 Claims, 34 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 30

☐ 12. Document ID: US 5953530 A

L24: Entry 12 of 13 File: USPT Sep 14, 1999

US-PAT-NO: 5953530
DOCUMENT-IDENTIFIER: US 5953530 A
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for run-time memory access checking and memory leak detection of a multi-threaded program

DATE-ISSUED: September 14, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rishi; Alok	El Granada	CA		
Masamitsu; Jon A.	Livermore	CA		

US-CL-CURRENT: 717/127; 714/38, 714/48

ABSTRACT:

The present invention is a system and method for a "debugger Run-Time-Checking for valid memory accesses for multi-threaded application programs" (hereinafter "RTC/MT") wherein a run-time process which includes multiple threads running either serially or concurrently, may be monitored by a debugger program and memory access errors detected

and correctly attributed to the process thread encountering the error. The RTC/MT system of the present invention also provides an apparatus and method which monitors and reports memory leaks as required for multi-threaded target programs.

25 Claims, 8 Drawing figures
Exemplary Claim Number: 18
Number of Drawing Sheets: 8

☐ 13. Document ID: US 5179702 A

L24: Entry 13 of 13 File: USPT Jan 12, 1993

US-PAT-NO: 5179702
DOCUMENT-IDENTIFIER: US 5179702 A

TITLE: System and method for controlling a highly parallel multiprocessor using an anarchy based scheduler for parallel execution thread scheduling

DATE-ISSUED: January 12, 1993

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Spix; George A.	Eau Claire	WI		
Wengelski; Diane M.	Eau Claire	WI		
Hawkinson; Stuart W.	Eau Claire	WI		
Johnson; Mark D.	Eau Claire	WI		
Burke; Jeremiah D.	Eau Claire	WI		
Thompson; Keith J.	Eau Claire	WI		
Gaertner; Gregory G.	Eau Claire	WI		
Brussino; Giacomo G.	Eau Claire	WI		
Hessel; Richard E.	Altoona	WI		
Barkai; David M.	Eau Claire	WI		
Chen; Steve S.	Chippewa Falls	WI		
Oslon; Steven G.	Chippewa Falls	WI		
Strout, II; Robert E.	Livermore	CA		
Masamitsu; Jon A.	Livermore	CA		
Cox; David M.	Livermore	CA		
O'Gara; Linda J.	Livermore	CA		
O'Hair; Kelly T.	Livermore	CA		
Seberger; David A.	Livermore	CA		
Rasbold; James C.	Livermore	CA		
Cramer; Timothy J.	Pleasanton	CA		
Van Dyke; Don A.	Pleasanton	CA		
Chandramouli; Ashok	Fremont	CA		

US-CL-CURRENT: 718/102; 717/124, 717/146, 717/151, 718/104, 718/106

ABSTRACT:

An integrated software architecture for a highly parallel multiprocessor system having multiple tightly-coupled processors that share a common memory efficiently controls the interface with and execution of programs on such a multiprocessor system. The software architecture combines a symmetrically integrated multithreaded operating system and an integrated parallel user environment. The operating system distributively implements an anarchy-based scheduling model for the scheduling of processes and resources by allowing each processor to access a single image of the operating system stored in the common memory that operates on a common set of operating system shared resources. The user environment provides a common visual representation for a plurality of program development tools that provide compilation, execution and debugging capabilities for multithreaded user programs and assumes parallelism as the standard mode of operation.

12 Claims, 60 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 53

Terms	Documents
L23 and (context near (switch or swap))	13